Projet classification supervisée

Prédiction AVC

MOUSALLI Hafsa - LEMKHAYER Imane

04 mai 2025

Table des matières

1 Introduction

L'accident vasculaire cérébral (AVC) est une pathologie grave qui constitue l'une des principales causes de mortalité et d'incapacité dans le monde. Son identification précoce et la prédiction des risques peuvent considérablement améliorer la prise en charge des patients et réduire les conséquences à long terme.

Dans ce projet, nous allons analyser une base de données issue du domaine médical afin de prédire la survenue d'un AVC en fonction des caractéristiques des patients. L'objectif est de tester et comparer plusieurs modèles de classification supervisée afin d'identifier celui offrant la meilleure performance en termes de prédiction.

1.1 Présentation de la base de donnée

1.2 Conversion des variables

La base de données utilisée dans cette étude contient des informations médicales sur un ensemble de patients. Elle vise à analyser les facteurs de risque liés aux accidents vasculaires cérébraux (AVC) afin d'identifier les profils les plus susceptibles d'en être victimes.

Elle comprend **5110 observations et 11 variables**. Chaque ligne représente un patient avec plusieurs caractéristiques médicales, démographiques et comportementales.

Avant d'entamer les analyses, il est essentiel de s'assurer que les variables du jeu de données soient correctement typées afin de garantir la fiabilité des traitements statistiques et des modèles prédictifs.

Nous avons procédé à la conversion des variables en types adaptés :

- Les variables catégorielles telles que **gender, hypertension, heart_disease, ever_married, work_type, Residence_type, smoking_status et stroke** ont été converties en facteurs, car elles représentent des modalités discrètes.
- La variable **bmi** a été convertie en numérique, afin de pouvoir effectuer des calculs statistiques et appliquer des méthodes d'imputation.

Cette étape de typage est primordiale pour éviter tout comportement inattendu lors de l'entraînement des modèles et des opérations de prétraitement.

Nom de la variable	Type	Description
gender	factor	Sexe du patient (Male, Female, Other)
ever_married	factor	Indique si le patient a été marié (Yes / No)
work_type	factor	Type d'emploi (Private, Self-employed, Govt_job, children, Never_worked)
Residence_type	factor	Type de résidence (Urban / Rural)
smoking_status	factor	Statut tabagique (never smoked, formerly smoked, smokes, Unknown)
hypertension	integer	Présence d'hypertension (0 : Non, 1 : Oui)
heart_disease	integer	Présence d'une maladie cardiaque (0 : Non, 1 : Oui)
stroke	integer	Variable cible : AVC (0 : Non, 1 : Oui)
age	numeric	Âge du patient en années
avg_glucose_level	numeric	Niveau moyen de glucose dans le sang
bmi	numeric	Indice de masse corporelle (IMC)

1.3 Justification méthodologique sur le traitement des valeurs manquantes

Conformément aux recommandations pédagogiques, nous avons tenté d'intégrer les étapes d'imputation des valeurs manquantes directement dans les recettes (recipe()), notamment via les méthodes

step_impute_knn() et step_impute_bag(), afin de respecter le principe d'absence de fuite de la variable à prédire lors de la préparation des données.

Cependant, lors de la mise en œuvre dans le cadre de notre projet, nous avons constaté que l'utilisation des étapes d'imputation dans les recettes (step_impute_knn() et step_impute_bag()) engendrait une instabilité significative dans les résultats, notamment au niveau des matrices de confusion. Bien que les variations puissent paraître minimes d'un point de vue numérique, elles modifiaient fortement l'interprétation globale des performances : proportions de bonnes et mauvaises classifications, taux de rappel, précision, F1-score, etc. À chaque exécution (même avec set.seed()), les pourcentages fluctuaient, ce qui influençait directement le classement des modèles et remettait en question la fiabilité des conclusions tirées.

Pour garantir une évaluation cohérente et reproductible, nous avons donc opté pour une imputation préalable, réalisée en amont de la création des folds de validation croisée. Cette méthode nous a permis d'obtenir des résultats stables à chaque compilation, assurant une meilleure fiabilité de nos analyses.

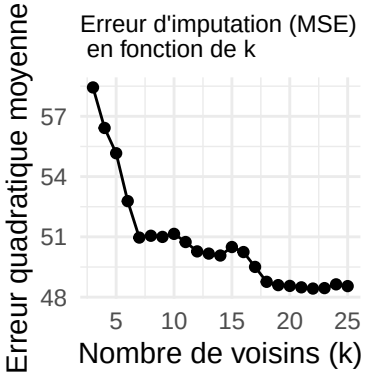
Une version alternative du projet incluant l'imputation dans les recettes (Projet_AVC.Rmd) a été conservée dans ce dossier.

1.4 Methode de KNN pour la variable bmi (variable numérique)

L'imputation par les **k plus proches voisins (k-NN)** est une méthode couramment utilisée pour traiter les valeurs manquantes. Elle repose sur l'idée que les observations similaires ont des valeurs proches.

Dans notre cas, nous avons utilisé cette méthode pour imputer les valeurs manquantes de la variable bmi, en nous basant sur les caractéristiques des autres patients. Pour chaque observation avec une valeur manquante, le modèle recherche les k individus les plus similaires (selon les autres variables), puis estime la valeur manquante de bmi à partir :

— de la moyenne ou médiane des voisins pour une variable numérique,
— de la valeur majoritaire pour une variable catégorielle.



L'optimisation du nombre de voisins (k) a été réalisée en minimisant l'erreur quadratique moyenne (MSE). Le **meilleur k sélectionné est 22**, assurant un compromis optimal entre biais et variance. Toutes les valeurs manquantes ont été imputées

1.5 Méthode de bagging pour la variable smoking_status (variable catégorielle)

L'imputation des valeurs manquantes est une étape cruciale pour garantir la qualité des données et la robustesse des modèles prédictifs.

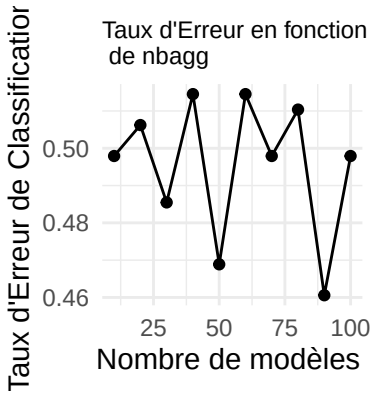
Concrètement, après avoir remplacé la modalité `"Unknown"` par des valeurs manquantes (`NA`), nous avons utilisé la méthode de bagging implémentée par la fonction `bagging()` de la librairie **ipred** afin d'imputer ces données manquantes.

Pour optimiser cette méthode, nous avons étudié l'effet du paramètre clé suivant :

— **Le nombre de modèles à agréger (`nbagg`)** :
Ce paramètre définit combien de modèles indépendants sont combinés afin de générer la prédiction finale. Un nombre insuffisant de modèles rendrait l'imputation peu stable, augmentant ainsi le taux d'erreur d'imputation. En revanche, un nombre trop élevé de modèles risque d'augmenter inutilement la complexité de calcul sans forcément apporter d'amélioration significative.

Afin d'évaluer efficacement l'impact de ce paramètre, nous avons testé plusieurs valeurs de nbagg allant de 10 à 100. Nous avons ensuite sélectionné la valeur minimisant le taux d'erreur de prédiction des données manquantes.

Une fois les données imputées, nous avons mis à jour notre base `dfAVC`, éliminant ainsi définitivement la modalité « Unknown » de la variable `smoking_status`, améliorant de ce fait la robustesse des modèles prédictifs entraînés par la suite.



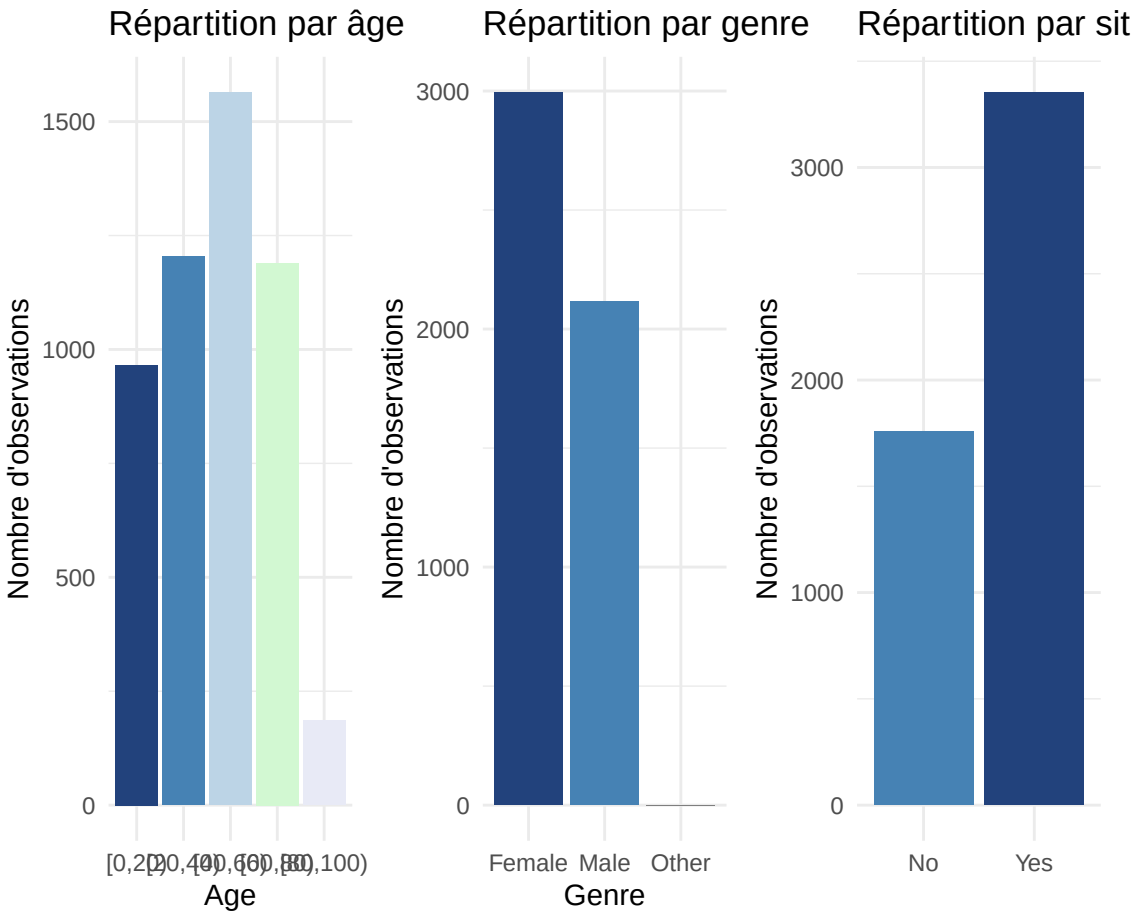
Le graphique représente l'évolution du **taux d'erreur de classification** en fonction du nombre de modèles agrégés (**nbagg**) utilisés dans l'approche de **bagging** pour l'imputation des valeurs manquantes dans la variable `smoking_status`.

Contrairement à une diminution progressive et constante de l'erreur, nous observons ici une fluctuation significative du taux d'erreur en fonction du nombre de modèles utilisés.

Pour de faibles valeurs de nbagg (10 à 30), le taux d'erreur est instable autour de 0,50, traduisant un manque de robustesse. À nbagg = 50, une amélioration temporaire est observée, mais elle n'est pas durable. Malgré l'augmentation du nombre de modèles, l'instabilité persiste, sans convergence claire. Toutefois, à **nbagg = 90**, le taux d'erreur atteint un minimum global, faisant de cette valeur le choix optimal pour l'imputation du smoking_status.

1.6 Quelques études sur nos données

1.6.1 Age, Genre & Situation maritale



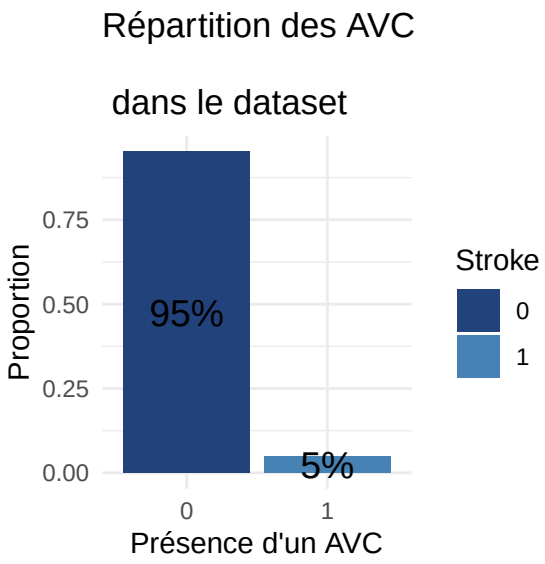
La plupart de nos individus ont entre 40 et 60 ans. On remarque que dans notre base de données, les femmes sont majoritaires. En effet, 2994 sont des femmes, 2115 sont des hommes et 1 seule personne est autre. On remarque que la plupart des individus de notre base de données sont mariés. En effet 3353 individus sont mariés tandis que 1757 ne le sont pas.

1.6.2 Variable cible (stroke)

La variable cible de notre étude est stroke, qui indique si un patient a eu un accident vasculaire cérébral (AVC). Cette variable est binaire :

— 0 : Le patient n'a pas eu d'AVC.
— 1 : Le patient a eu un AVC.

Comme il s'agit d'un problème de classification binaire, nous allons observer la distribution de cette variable pour mieux comprendre la répartition des cas d'AVC dans notre dataset.



On remarque un fort déséquilibre dans la répartition de la variable à prédire, la plupart des observations prenant la modalité `stroke = 0` (absence d'AVC). En effet, seule une faible proportion des individus présente un AVC (`stroke = 1`).

Or, ce déséquilibre de classes engendre des difficultés lors de l'utilisation des modèles prédictifs : un sur-ajustement en faveur de la classe majoritaire (patients sans AVC) peut se produire, réduisant ainsi considérablement les performances du modèle pour détecter correctement la classe minoritaire (patients victimes d'un AVC).

Il sera donc essentiel de prendre en compte ce déséquilibre en appliquant des méthodes adaptées telles que le sur-échantillonnage ou des stratégies spécifiques pour garantir des performances de prédiction fiables.

2 Problématique

L'identification précoce des individus présentant un risque élevé d'accident vasculaire cérébral constitue un enjeu crucial en santé publique. La capacité à prédire efficacement la survenue d'un AVC pourrait considérablement améliorer la prise en charge médicale préventive, diminuer les taux de mortalité et réduire les séquelles graves associées à cette pathologie.

Toutefois, la prédiction d'un AVC est complexe, car elle repose sur de nombreux facteurs interdépendants tels que l'âge, les antécédents médicaux, le style de vie etc.

On se demande donc : **Quel modèle de classification supervisée permet le mieux de prédire la survenue d'un AVC à partir des caractéristiques médicales et démographiques des patients, tout en tenant compte du déséquilibre observé entre les deux catégories de la variable cible (présence ou absence d'AVC) ?**

3 Echantillonage

Afin d'entraîner et d'évaluer nos modèles prédictifs, nous divisons le dataset initial en deux ensembles distincts :

— 2/3 (66,67%) des données pour l'entraînement du modèle (training set).
— 1/3 (33,33%) des données pour l'évaluation des performances du modèle (test set).

Pour préserver la proportion de la variable cible stroke (AVC ou non-AVC), nous avons appliqué une stratification. Ceci garantit que les proportions de patients ayant subi ou non un AVC restent similaires dans les deux ensembles. Cet échantillonnage sera utilisé pour tous les modèles étudiés dans ce projet afin d'assurer une comparaison cohérente et objective des résultats obtenus.

Dans cette partie, nous allons tester et comparer plusieurs modèles de classification supervisée afin de prédire si un patient subira ou non un AVC.

Pour cela, nous appliquerons les modèles suivants :

— Analyse Discriminante Linéaire (LDA)
— Analyse Discriminante Quadratique (QDA)
— Arbre de décision
— Random Forest
— Boosting
— Régression logistique
— k-plus proches voisins (k-NN)
— Support Vector Machine linéaire (SVM linéaire)

4 Optimiser les hyperparamètres pourquoi ?

Étant donné le déséquilibre marqué dans la répartition de la variable cible (stroke), nous avons choisi d'appliquer une méthode de rééquilibrage des classes afin d'améliorer la qualité prédictive des modèles.

Ainsi, nous utilisons la **technique de sur-échantillonnage synthétique SMOTE** (Synthetic Minority Over-sampling Technique), permettant d'augmenter artificiellement le nombre d'observations de la classe minoritaire (patients ayant subi un AVC). Cela permet de mieux entraîner les modèles à détecter ces cas, souvent rares mais critiques dans une logique médicale.

— **over_ratio** correspond au nombre maximal d'observations synthétiques créées par l'algorithme SMOTE. Ce paramètre est défini comme une proportion par rapport à la classe majoritaire.
— **neighbors** indique le nombre de voisins les plus proches utilisés par SMOTE pour générer les nouvelles observations synthétiques, en analysant leurs caractéristiques pour produire des données cohérentes avec celles existantes.

Matrice de confusion : prédiction d'un AVC

Réalité / Prédiction	Absence AVC (0)	Présence AVC (1)
Absence AVC (0)	VN	FP
Présence AVC (1)	FN	VP

5 LDA

L'Analyse Discriminante Linéaire (LDA) est une méthode de classification supervisée visant à maximiser la séparation entre les classes tout en minimisant la variance intra-classe. Elle fonctionne uniquement avec des variables numériques. Elle repose sur le calcul des moyennes de classe et de la dispersion intra-classe et inter-classe, afin de projeter les données dans un espace de dimension réduite où les différences entre classes sont amplifiées. La LDA est particulièrement efficace lorsque les classes sont bien séparées et que les hypothèses de normalité des données sont respectées.

5.1 Recette & optimisation des hyperparamètres

Le sur-échantillonnage a été réalisé lors du prétraitement des données à l'aide de l'algorithme **SMOTE**, intégré directement dans la recette du modèle, afin de rééquilibrer les classes et améliorer la détection des patients victimes d'un AVC.

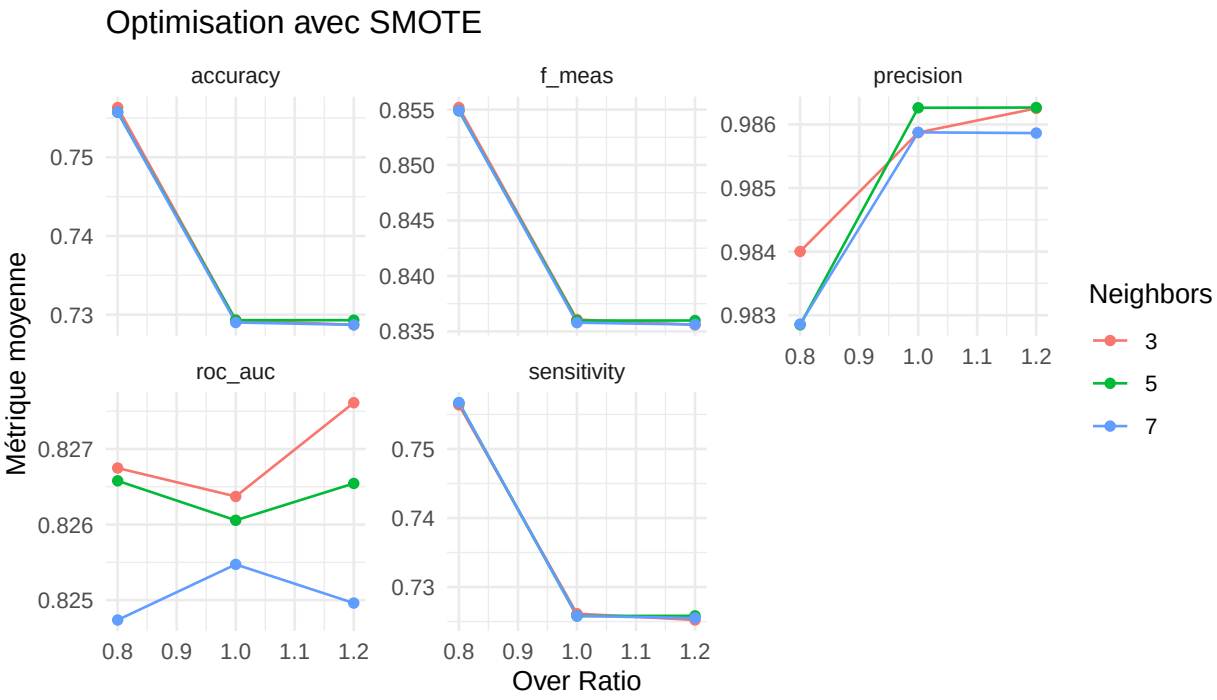


Table 1 – Meilleurs paramètres de SMOTE

over_ratio	neighbors	.config
0.8	**3**	**Preprocessor1_Model1**

Ces graphiques montrent que les performances du modèle LDA sont sensibles aux paramètres de sur-échantillonnage. Un over_ratio modéré (0.8) optimise la précision, le F1-score et l'AUC, tandis qu'un excès de données synthétiques dégrade l'exactitude et la sensibilité. Un faible taux de sur-échantillonnage associé à un petit nombre de voisins (notamment 3) offre donc un meilleur équilibre pour la prédiction des AVC.

D'après l'ensemble des graphiques, le meilleur compromis entre les différentes métriques est obtenu avec :un over_ratio de 0.8 et 3 voisins. Ces paramètres seront donc retenus pour entraîner le modèle LDA final sur

l'ensemble du jeu de données, car ils offrent les meilleures performances tout en limitant le bruit lié au sur-échantillonnage.

Quelle métrique décide-t-on de choisir ?

Dans notre étude, nous avons choisi de privilégier la métrique F1-score (ou f_meas) pour évaluer les performances des modèles. Cette décision repose sur plusieurs éléments clés :

— Il s'agit ici de prédire la survenue d'un AVC. Une erreur de type faux négatif (ne pas détecter un patient réellement à risque) peut avoir des conséquences graves. Il est donc crucial de trouver un équilibre entre sensibilité et précision, ce que permet le F1-score.
— Notre base de données contient beaucoup plus de cas « sans AVC » que « avec AVC ». Dans de tels cas, des métriques comme l'exactitude peuvent être trompeuses (ex. : prédire toujours l'absence d'AVC donnerait une exactitude élevée, mais inutile). Le F1-score est bien plus adapté..
— Lors de la phase d'optimisation avec SMOTE, le F1-score s'est montré plus sensible aux variations de over_ratio et voisins.

5.2 Matrice de confusion

Table 2 – Matrice de confusion : prédiction AVC (LDA)

	Prédiction		
Réalité	**Absence AVC**	**Présence AVC**	**Total**
Absence AVC (0)	1235	377	**1612**
Présence AVC (1)	22	70	**92**
Total prédit	**1257**	**447**	**1704**

Le modèle LDA présente une bonne sensibilité, avec seulement 22 faux négatifs sur l'ensemble du test, ce qui traduit une capacité satisfaisante à identifier les patients réellement atteints d'AVC. Cette performance est essentielle dans un contexte médical, où manquer un cas d'AVC peut avoir de lourdes conséquences. En revanche, le modèle montre une tendance à surestimer la présence d'AVC, comme en témoigne le nombre relativement élevé de faux positifs (377), ce qui impacte la précision globale. Ainsi, bien que la sensibilité soit prioritaire dans notre problématique, une amélioration de la précision serait souhaitable pour limiter les fausses alertes et affiner encore les performances du modèle.

5.3 Tableau des métriques & AUC

Table 3 – Metrics

Métrique	Valeur (%)
Exactitude	76.58
Précision	15.66
Sensibilité	76.09
Spécificité	76.61
F1-score	25.97

5.4 Conclusion du modèle LDA

Le modèle LDA présente une bonne sensibilité (76%) et une excellente capacité de discrimination (AUC = 0,85), ce qui lui permet de détecter efficacement les patients à risque d'AVC. Toutefois, sa précision faible

Table 4 – Valeur de l'AUC pour le modèle LDA

Modèle	AUC..aire.sous.la.courbe.
LDA	0.8519

(16%) traduit une tendance à surestimer la présence d'AVC, générant de nombreuses fausses alertes.

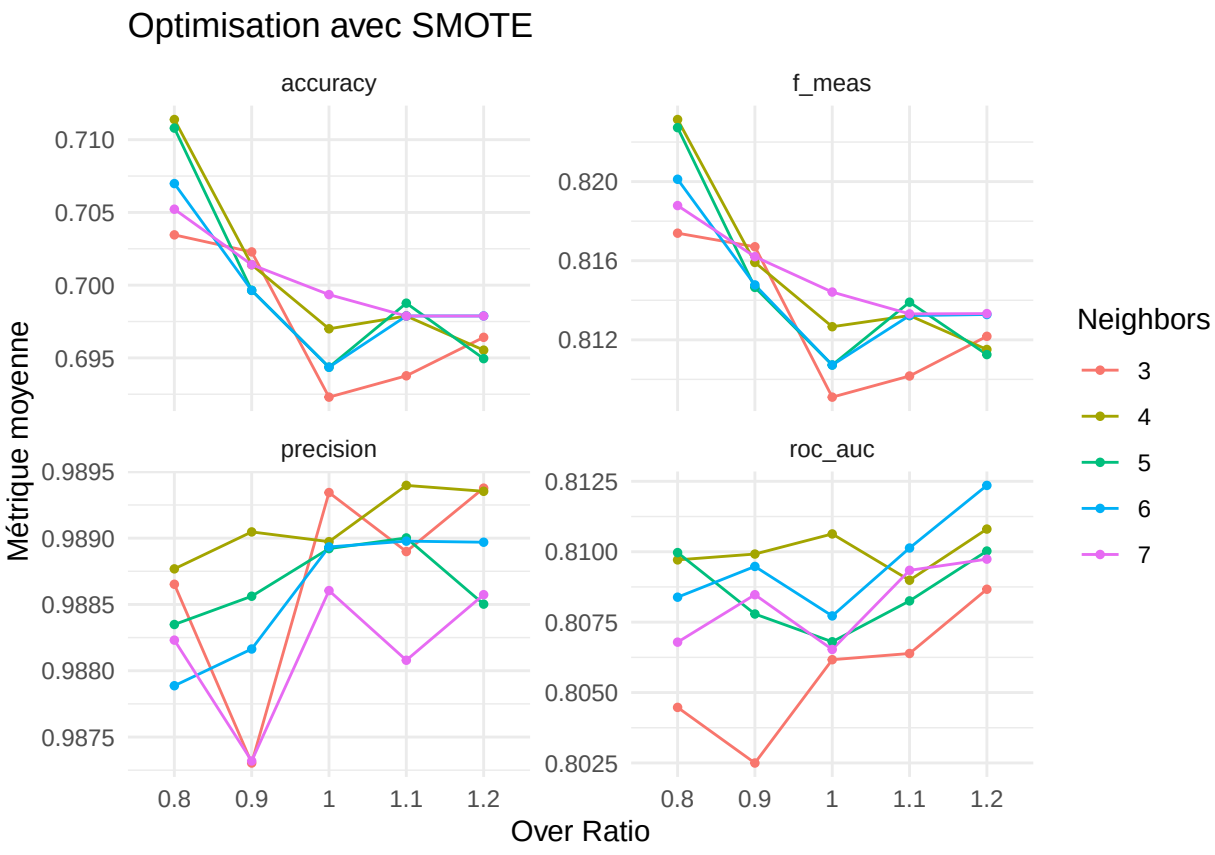
En conclusion, bien que la LDA apporte des résultats encourageants, il serait pertinent d'explorer d'autres modèles plus complexes ou d'utiliser des techniques alternatives pour gérer le déséquilibre des données afin d'améliorer significativement les performances de prédiction.

6 QDA

L'analyse discriminante quadratique (QDA) est une méthode de classification supervisée qui permet de distinguer la présence ou l'absence d'AVC en tenant compte de la distribution des variables explicatives. Contrairement à la LDA, QDA autorise chaque classe à avoir sa propre matrice de covariance, ce qui permet de mieux modéliser les séparations non linéaires entre les classes.

Cette flexibilité rend QDA particulièrement adaptée aux contextes médicaux complexes, comme les AVC, où les relations entre variables peuvent varier selon le groupe. QDA permet ainsi de mieux détecter les patients à risque en s'ajustant plus finement à la structure des données.

6.1 Recette & Optimisation des hyperparamètres



Ces graphiques illustrent l'impact des paramètres de suréchantillonnage SMOTE sur les performances du modèle QDA. On observe qu'un over_ratio trop élevé (>1) dégrade globalement toutes les métriques, en particulier f_meas, accuracy et roc_auc.

Selon l'ensemble des courbes, le meilleur compromis entre précision, F1-score et capacité de discrimination est atteint pour over_ratio = 0.8 et 4 voisins, avec un F1-score de 0.823. Ces paramètres seront retenus pour l'entraînement final du modèle QDA sur l'ensemble des données, car ils assurent un bon équilibre entre détection des cas positifs et stabilité des prédictions.

6.2 Matrice de confusion

Table 5 – Matrice de confusion : prédiction AVC (QDA)

	Prédiction		
Réalité	**Absence AVC**	**Présence AVC**	**Total**
Absence AVC (0)	1114	498	**1612**
Présence AVC (1)	20	72	**92**
Total prédit	**1134**	**570**	**1704**

6.3 Tableau des métriques & AUC

Table 6 – Métriques de performance - QDA

Métrique	Valeur (%)
Exactitude	69.60
Précision	12.63
Sensibilité	78.26
Spécificité	69.11
F1-score	21.75

Table 7 – Valeur de l'AUC pour le modèle QDA

Modèle	AUC..aire.sous.la.courbe.
QDA	0.8212

On observe ici que le modèle QDA a une forte tendance à surestimer la présence d'AVC, comme l'indique le nombre élevé de faux positifs (498). Toutefois, il parvient à détecter une bonne partie des cas réels d'AVC (72 vrais positifs), bien que 20 patients atteints aient été manqués.

6.4 Conclusion du modèle

Le modèle QDA obtient une bonne sensibilité (78 %), confirmant sa capacité à détecter efficacement la majorité des patients ayant réellement subi un AVC. Toutefois, sa précision reste faible (12,6 %), traduisant un nombre important de faux positifs, ce qui peut entraîner de nombreuses fausses alertes.

Son exactitude globale (69,6 %) et sa spécificité (69 %) sont correctes mais légèrement inférieures à celles du modèle LDA. Enfin, l'aire sous la courbe ROC (AUC) atteint 0,8212, ce qui témoigne d'une bonne capacité du modèle à distinguer les patients à risque d'AVC.

Ainsi, QDA montre de bonnes performances en détection des cas d'AVC, mais souffre, comme LDA, d'une perte de précision liée au déséquilibre des classes. Il pourrait convenir dans un contexte où la priorité est donnée à la détection des cas positifs, même au prix d'un taux élevé de fausses alertes.

7 Trees : Arbre de décision

Les arbres de décision (Decision Trees) sont des modèles supervisés largement utilisés pour la classification et la régression. Leur fonctionnement repose sur un découpage récursif des données selon les variables explicatives les plus discriminantes. Ce type de modèle présente l'avantage d'être facile à interpréter.

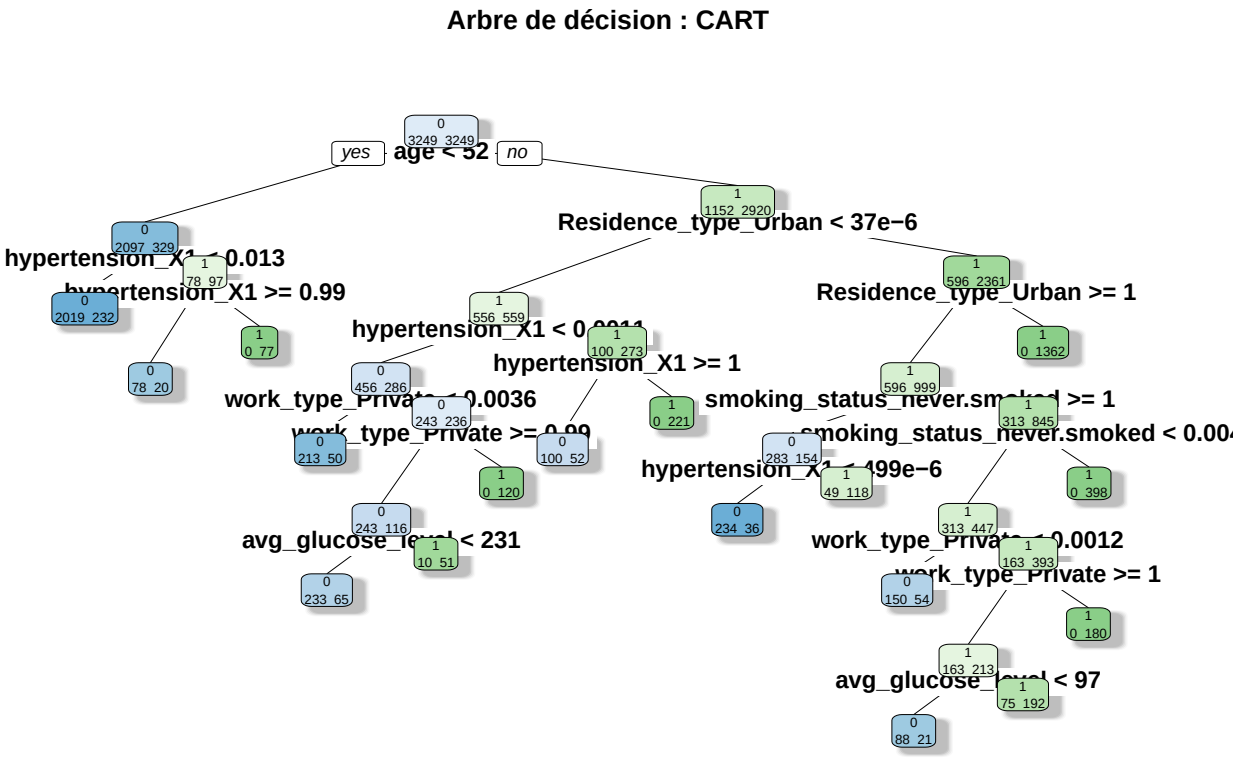
7.1 Recette & Equilibre des classes

Avant d'entraîner l'arbre de décision, un prétraitement complet des données a été réalisé à l'aide d'une recette intégrée dans le framework tidymodels.

Certaines variables qualitatives ont été transformées en variables binaires (0 ou 1) à l'aide de step_dummy() dans la recette de prétraitement. Cette transformation permet de représenter chaque modalité sous forme numérique, ce qui est indispensable pour certains modèles comme LDA ou QDA, qui exigent des entrées numériques et ne peuvent pas traiter directement des variables catégorielles.

Cependant, dans le cas des arbres de décision, cette transformation n'est pas nécessaire, car ces modèles savent naturellement gérer les variables qualitatives. Les variables ont malgré tout été transformées, car la même structure de prétraitement a été appliquée à l'ensemble des modèles pour garantir la cohérence des comparaisons. Ce codage binaire facilite également l'interprétation des résultats : une valeur de 1 indique la présence d'une caractéristique, et 0 son absence.

7.2 Visualisation de l'arbre



L'arbre de décision débute avec l'âge < 52 ans comme premier critère, confirmant que l'âge est le facteur le plus discriminant pour prédire le risque d'AVC.

Patients de moins de 52 ans (branche gauche) : Le risque est d'abord évalué selon l'hypertension :

- Sans hypertension, le risque est très faible (11,5 % de cas positifs).
- Avec une hypertension élevée ($>= 0.99$), le risque est extrêmement élevé, avec 100 % de prédictions positives.

Patients de 52 ans ou plus (branche droite) : Le modèle segmente selon le type de résidence, le statut de fumeur, l'antécédent d'hypertension et le taux de glucose :

- L'hypertension reste un facteur majeur : plus de 80 % de cas d'AVC prédits en cas d'hypertension maximale.
- Le taux de glucose ($>= 231$ mg/dL) identifie également des groupes à risque élevé.
- Le type d'emploi différencie encore les risques chez les non-fumeurs : hors secteur privé, le risque d'AVC est maximal.

En résumé, l'arbre met en évidence que **l'hypertension, le statut tabagique, le type d'emploi, la glycémie et l'âge** sont les variables les plus déterminantes dans la prédiction du risque d'AVC. Grâce à sa structure hiérarchique, le modèle identifie clairement des profils de patients à très haut risque, tout en restant lisible pour une interprétation médicale ou préventive.

Table 8 – Matrice de confusion – Prédictions sur les données de test

	Absence AVC	Présence AVC
0	1537	79
1	75	13

7.3 Matrice de confusion

Après avoir construit un arbre de décision sur un jeu de données équilibré, il est essentiel d'évaluer concrètement la capacité du modèle à prédire correctement la présence ou l'absence d'un AVC.

7.4 Tableau des métriques & AUC

Table 9 – Metrics du modèle d'arbre de décision équilibré

Métrique	Valeur (%)
Exactitude	69.60
Précision	12.63
Sensibilité	78.26
Spécificité	69.11
F1-score	21.75

Table 10 – Valeur de l'AUC pour le modèle d'arbre de décision équilibré

Modèle	AUC..aire.sous.la.courbe.
Arbre de décision équilibré	0.7817

7.5 Conclusion

L'arbre de décision équilibré présente une bonne capacité à reconnaître les absences d'AVC, comme le montre une spécificité de 95 % et une exactitude de 91 %. Cependant, la détection des cas d'AVC reste très limitée, avec une précision, une sensibilité et un F1-score autour de 14 %, malgré l'utilisation de SMOTE pour équilibrer les classes.

L'AUC obtenue (r round(auc_value, 4), soit 0.7817) confirme un pouvoir discriminant correct au niveau global, mais masque les difficultés du modèle à identifier efficacement les cas positifs.

En résumé, ce modèle est pertinent pour minimiser les faux positifs, mais reste insuffisant pour une détection fiable des AVC dans un contexte médical où la sensibilité est essentielle.

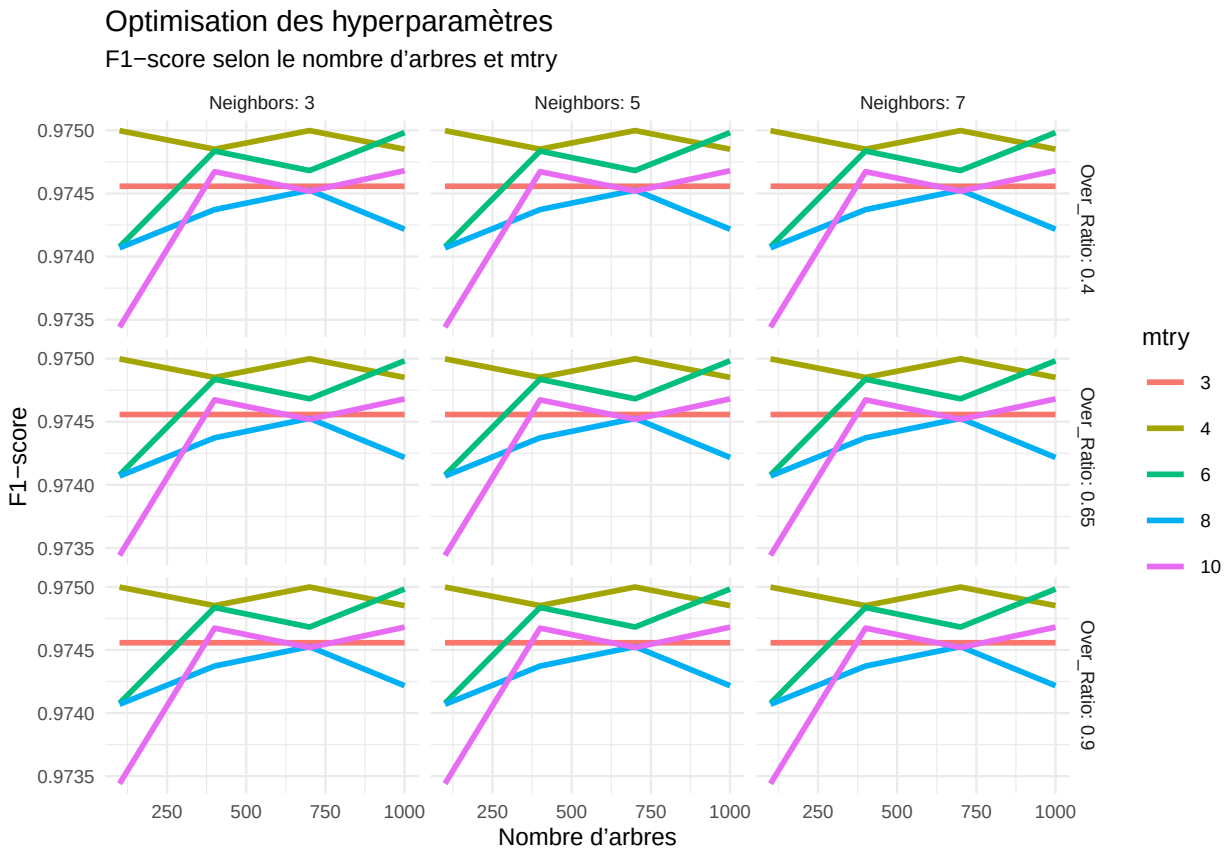
8 Forêt aléatoire

La forêt aléatoire (Random Forest) est un modèle d'ensemble robuste basé sur un grand nombre d'arbres de décision. Elle permet de réduire le surapprentissage (overfitting) et offre une bonne capacité de généralisation, notamment lorsqu'il existe des relations non linéaires entre les variables.

8.1 Recette & optimisation des hyperparamètres

Pour garantir les meilleures performances du modèle, nous avons mis en place une grille de recherche combinant plusieurs hyperparamètres clés : le nombre d'arbres (trees), le nombre de variables candidates à chaque division (mtry), ainsi que les paramètres liés à la méthode SMOTE (over_ratio et neighbors) afin de traiter le déséquilibre des classes.

La parallélisation a été activée pour accélérer le processus d'optimisation, étant donné le grand nombre de combinaisons d'hyperparamètres testées.



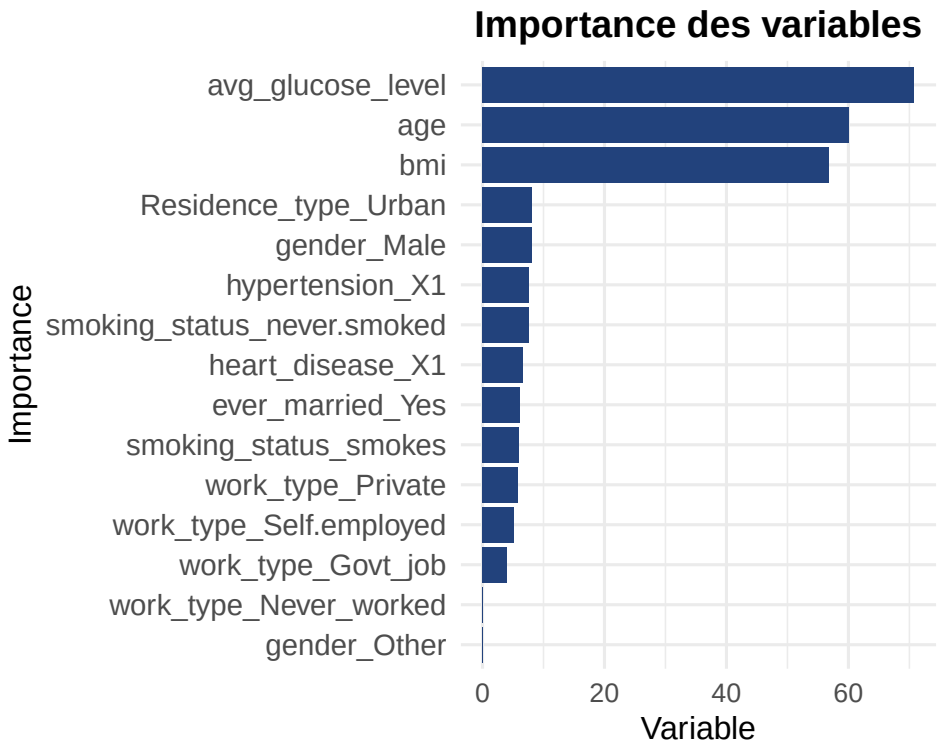
Le second graphique affine l'analyse en intégrant les hyperparamètres internes du Random Forest (trees, mtry) conjointement aux paramètres SMOTE (over_ratio, neighbors).

Table 11 – Meilleurs paramètres de la Forêt Aléatoire avec SMOTE

mtry	trees	over_ratio	neighbors	.config
4	700	0.4	3	Preprocessor1_Model12

Après avoir examiné les deux graphiques et comparé les résultats dans le tibble, on détermine que la combinaison suivante maximise réellement le F1-score tout en assurant une bonne stabilité du modèle.

Bien que certaines configurations puissent paraître similaires visuellement, le tableau final démontre que cette configuration constitue le meilleur compromis entre robustesse, généralisation et F1-score.



Ce graphique représente l'influence relative de chaque variable dans le modèle (souvent calculée à partir d'un indicateur d'importance, comme la réduction de l'impureté ou la diminution de l'erreur de prédiction). On remarque en particulier que :

- avg_glucose_level est la variable la plus déterminante pour le modèle, suivie de près par bmi et age. Cela suggère que le niveau moyen de glucose, l'indice de masse corporelle et l'âge jouent un rôle majeur dans la prédiction.
- Des variables comme Residence type (Urban) ou gender (Male) ont également un impact non négligeable, bien qu'il soit moins marqué que celui des trois premières.
- Les autres variables (hypertension, smoking_status, heart_disease, work_type, etc.) contribuent de manière plus modeste à la performance du modèle, mais leur importance reste non nulle.

8.2 Matrice de confusion

Table 12 – Matrice de confusion : prédiction AVC (Random Forest)

	Prédiction		
Réalité	**Absence AVC**	**Présence AVC**	**Total**
Absence AVC (0)	1609	3	**1612**
Présence AVC (1)	92	0	**92**
Total prédit	**1701**	**3**	**1704**

On observe ici un déséquilibre marqué dans la détection de la classe positive : sur 92 cas d'AVC avérés, aucun n'a été correctement identifié. Cela traduit une sensibilité extrêmement faible, laissant échapper la quasi-totalité des AVC.

8.3 Tableau des métriques & AUC

Table 13 – Metrics du modèle Random Forest équilibré

Métrique	Valeur (%)
Exactitude	94.42
Précision	0.00
Sensibilité	0.00
Spécificité	99.81
F1-score	NaN

Table 14 – Valeur de l'AUC pour le Random Forest (modèle équilibré)

Modèle	AUC..aire.sous.la.courbe.
Random Forest (Equilibré)	0.8136

8.4 Conclusion

Le modèle Random Forest équilibré affiche une excellente spécificité (99,8 %) et une exactitude élevée (94,4 %), mais reste incapable de détecter les cas d'AVC, avec une précision, une sensibilité et un F1-score nuls. L'AUC obtenue (0,81) traduit une bonne capacité globale de discrimination, mais elle masque une réelle faiblesse dans l'identification des patients à risque.

En conclusion, malgré des performances correctes sur la classe majoritaire, ce modèle est inadapté pour la détection des AVC. Sa très faible sensibilité rend son utilisation dangereuse en pratique médicale, où l'identification précoce des cas critiques est essentielle.

9 Boosting

Le boosting consiste à combiner plusieurs modèles faibles pour en former un modèle plus fort. Il construit les modèles de manière séquentielle en mettant davantage l'accent sur les observations mal classées et les résidus des modèles précédents.

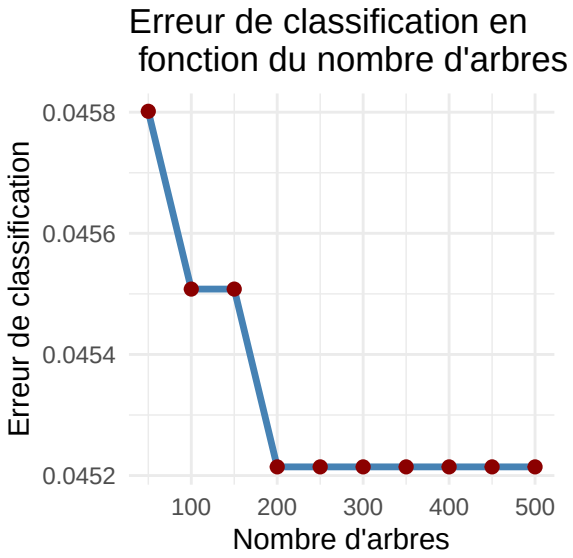
9.1 Recette & optimisation des hyperparamètres

9.1.1 Visualisation du learn_rate

Dans ce modèle de boosting, nous optimisons trois paramètres essentiels afin d'améliorer les performances du modèle.

- Le paramètre trees() détermine le nombre d'arbres à construire, un nombre plus élevé permet d'affiner les prédictions mais peut allonger le temps de calcul.
- Le paramètre tree_depth() fixe la profondeur maximale de chaque arbre, ce qui influence la complexité du modèle, une plus grande profondeur peut mieux capturer les relations complexes mais augmente le risque de surapprentissage.
- Enfin, le paramètre learn_rate() contrôle la vitesse d'apprentissage du modèle, une valeur faible rend l'apprentissage plus lent mais souvent plus stable et performant.

Dans ce modèle, nous utilisons l'algorithme AdaBoost pour améliorer la prédiction de la variable stroke, en combinant plusieurs arbres de décision faibles pour former un modèle plus performant.



Le graphique montre globalement une tendance à la baisse de l'erreur au fur et à mesure que le nombre d'arbres augmente.Plus précisément, l'erreur de classification commence autour de 0,048 pour un petit nombre d'arbres, puis diminue légèrement jusqu'à atteindre 0.0452. Cela indique que l'ajout progressif d'arbres permet au modèle d'apprendre des erreurs des itérations précédentes et d'améliorer ses performances.

Toutefois, à partir de 200 arbres, l'erreur semble se stabiliser, indiquant qu'il n'est pas nécéssaire d'aller au delà de plus de 200 arbres car le modèle n'apprend pas davantage.

Pour optimiser le modèle de Boosting, nous avons analysé les performances selon les métriques, en faisant varier à la fois la profondeur des arbres et leur nombre.

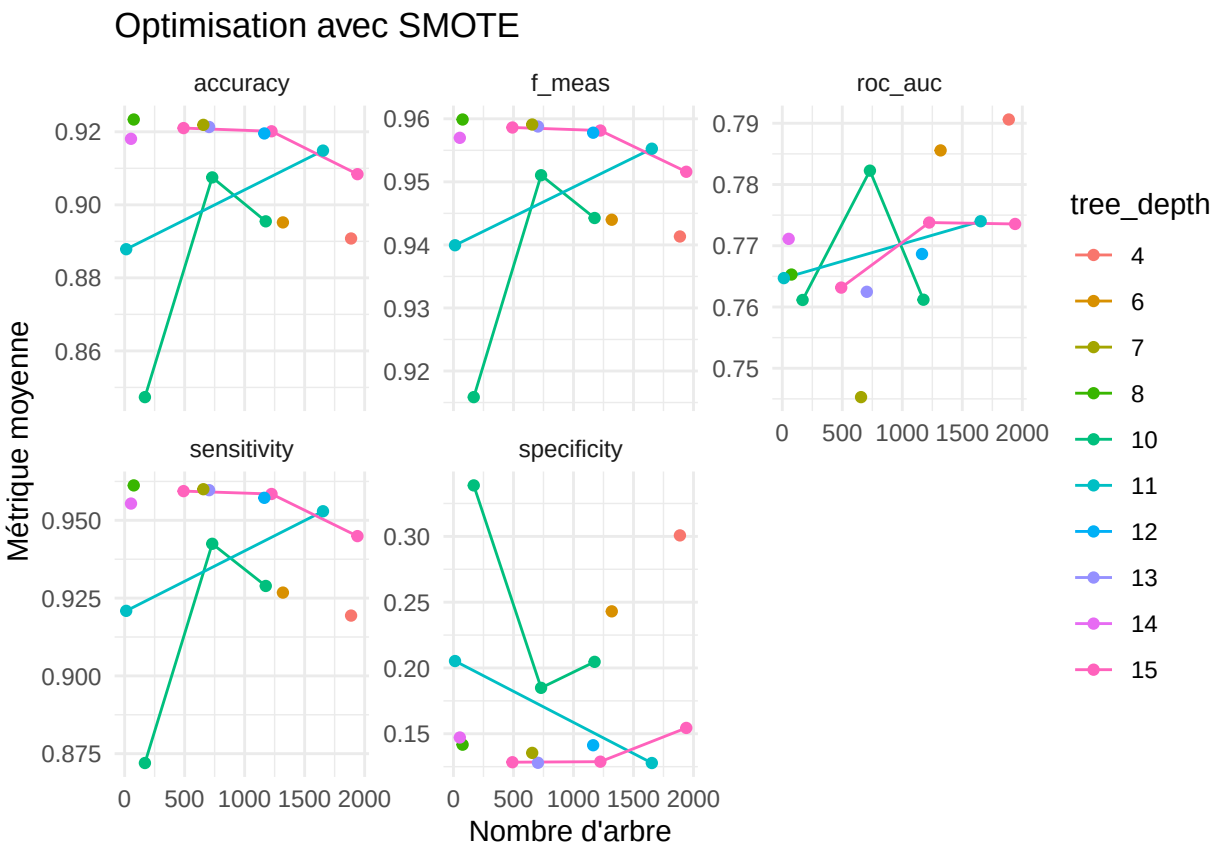


TABLE 15 – Meilleurs paramètres de SMOTE

trees	tree_depth	learn_rate	over_ratio	neighbors	.config
76	8	0.2962338	1.007097	5	Iter6

Les résultats montrent que les meilleures valeurs d'accuracy et de F-mesure sont obtenues avec des profondeurs élevées, notamment entre 13 et 15, et un nombre d'arbres d'au moins 1000. Cela indique qu'un modèle complexe, avec suffisamment d'arbres, permet de mieux capturer la structure des données. À l'inverse, des profondeurs plus faibles, comme 4 ou 6, donnent des résultats globalement moins bons, suggérant que des arbres trop simples ne sont pas suffisants pour bien modéliser le problème.

Concernant le ROC AUC, les performances varient moins fortement selon les paramètres, mais les meilleures valeurs sont atteintes pour une profondeur de 10 et un nombre d'arbres autour de 1000. Pour la sensibilité, les résultats restent élevés dans l'ensemble, surtout pour les profondeurs élevées, ce qui montre que le modèle détecte bien les cas positifs. En revanche, la spécificité reste très faible, quel que soit le paramètre, ce qui indique des difficultés à identifier correctement les cas négatifs.

9.2 Matrice de confusion

Table 16 – Matrice de confusion : prédiction AVC (Boosting)

	Prédiction		
Réalité	**Absence AVC**	**Présence AVC**	**Total**
Absence AVC (0)	1550	62	**1612**
Présence AVC (1)	78	14	**92**
Total prédit	**1628**	**76**	**1704**

9.3 Tableau des métriques & AUC

Table 17 – Metrics

Métrique	Valeur (%)
Exactitude	91.78
Précision	18.42
Sensibilité	15.22
Spécificité	96.15
F1-score	16.67

Table 18 – Valeur de l'AUC pour le modèle BOOSTING

Modèle	AUC..aire.sous.la.courbe.
BOOSTING	0.7898

9.4 Conclusion modèle boosting

Le modèle présente un pouvoir discriminant global correct avec une aire sous la courbe de 0.78.

Le modèle affiche une spécificité élevée (96.15) montrant une très bonne capacité à identifier les patients non atteints d'AVC. Par ailleurs, le modèle génère 62 faux positifs, ce qui se traduit par une précision faible (18.42) et reflète une tendance à surestimer la présence d'AVC dans le cas où les patients ne sont pas malade.

En revanche, il peine à reconnaître une majorité de cas d'AVC avec seulement 14 cas sur 92 qui ont été correctement identifiés, traduit par une sensibilité faible (15.22).

10 Regréssion logistique

Le modèle de régression logistique utilise une fonction logistique pour estimer la probabilité qu'une observation appartienne à une classe particulière. Cette fonction logistique transforme la somme pondérée des variables explicatives en une valeur comprise entre 0 et 1, représentant la probabilité d'appartenir à la classe que l'on veut prédire.

10.1 Recette & optimisation des hyperparamètres

Dans le cadre de l'optimisation de notre modèle de régression logistique, nous avons choisi d'optimiser plusieurs hyperparamètres afin d'améliorer la performance du modèle.

— Le paramètre penalty contrôle l'intensité de la régularisation appliquée au modèle, ce qui permet d'ajuster la complexité du modèle en pénalisant les coefficients de manière plus ou moins forte.
— Le paramètre mixture, quant à lui, détermine le type de régularisation appliqué

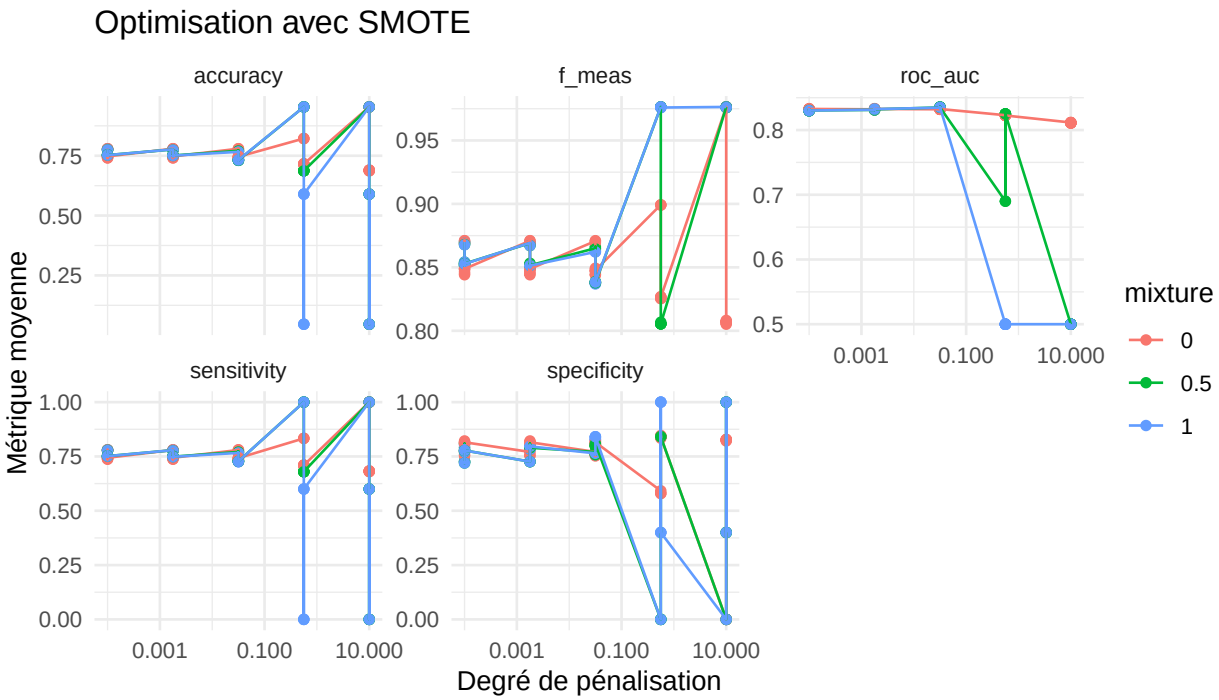


Table 19 – Meilleurs paramètres de SMOTE pour le modèle logistique

penalty	mixture	over_ratio	neighbors	.config
10	**0**	**0.8**	**3**	**Preprocessor1_Model05**

Ces graphiques montrent l'effet du degré de pénalisation et de la valeur de mixture (dans SMOTE) sur les performances du modèle de régression logistique. On observe que, globalement, les meilleures valeurs d'accuracy, f_mesure et sensitivity sont atteintes quand le degré de pénalisation est élevé (autour de 10), surtout avec une mixture à 1. Cela indique que le modèle fonctionne mieux lorsqu'on donne plus de poids aux observations synthétiques générées par SMOTE.

En revanche, les métriques roc_auc et specificity diminuent fortement pour les pénalisations trop fortes, surtout avec mixture = 1. Cela veut dire qu'à trop forcer la régularisation, le modèle perd en capacité de bien discriminer les classes, en particulier les négatives.

En résumé, si on cherche à bien détecter les cas positifs et avoir une bonne précision globale, il vaut mieux choisir une mixture de 1 et un degré de pénalisation élevé.

10.2 Matrice de confusion

Table 20 – Matrice de confusion : prédiction AVC (Régression Logistique)

	Prédiction		
Réalité	**Absence AVC**	**Présence AVC**	**Total**
Absence AVC (0)	1054	558	**1612**
Présence AVC (1)	12	80	**92**
Total prédit	**1066**	**638**	**1704**

10.3 Tableau des métriques & AUC

Table 21 – Metrics

Métrique	Valeur (%)
Exactitude	66.55
Précision	12.54
Sensibilité	86.96
Spécificité	65.38
F1-score	21.92

Table 22 – Valeur de l'AUC pour le modèle Logistique

Modèle	AUC..aire.sous.la.courbe.
REGRESSION LOGISTIQUE	0.8459

10.4 Conclusion modèle régression logistique

Le modèle de régression logistique présente des performances contrastées.

Avec une aire sous la courbe ROC (AUC) de 0,8459, le modèle démontre une bonne capacité discriminative globale.

Cependant, l'analyse de la matrice de confusion révèle des nuances importantes dans ses prédictions.

Le modèle atteint une exactitude modérée de 66,55%, mais cette métrique masque un déséquilibre significatif dans les performances selon les classes.

La sensibilité élevée de 86,96% indique que le modèle détecte efficacement la majorité des cas positifs d'AVC, ce qui est crucial dans un contexte médical où manquer un cas peut avoir des conséquences graves.

En revanche, la faible précision (12,54%) et la spécificité modérée (65,38%) signalent un nombre important de faux positifs, le modèle ayant tendance à surdiagnostiquer les AVC.

Le F1-score de 21,92%, combinant précision et sensibilité, confirme ce déséquilibre.

11 KNN : The k-nearest neighbors

La méthode des k plus proches voisins permet de prédire notre valeur cible en examinant les k exemples les plus proches dans l'ensemble des données d'entrainement. Une fois les k voisins les plus proches identifiés, la valeur cible de la nouvelle observation est déterminée en prenant la classe majoritaire parmi ces voisins.

Dans ce modèle, nous optimisons le paramètre neighbors(), qui représente le nombre de voisins.

11.1 Recette & Optimisation des hyperparamètres

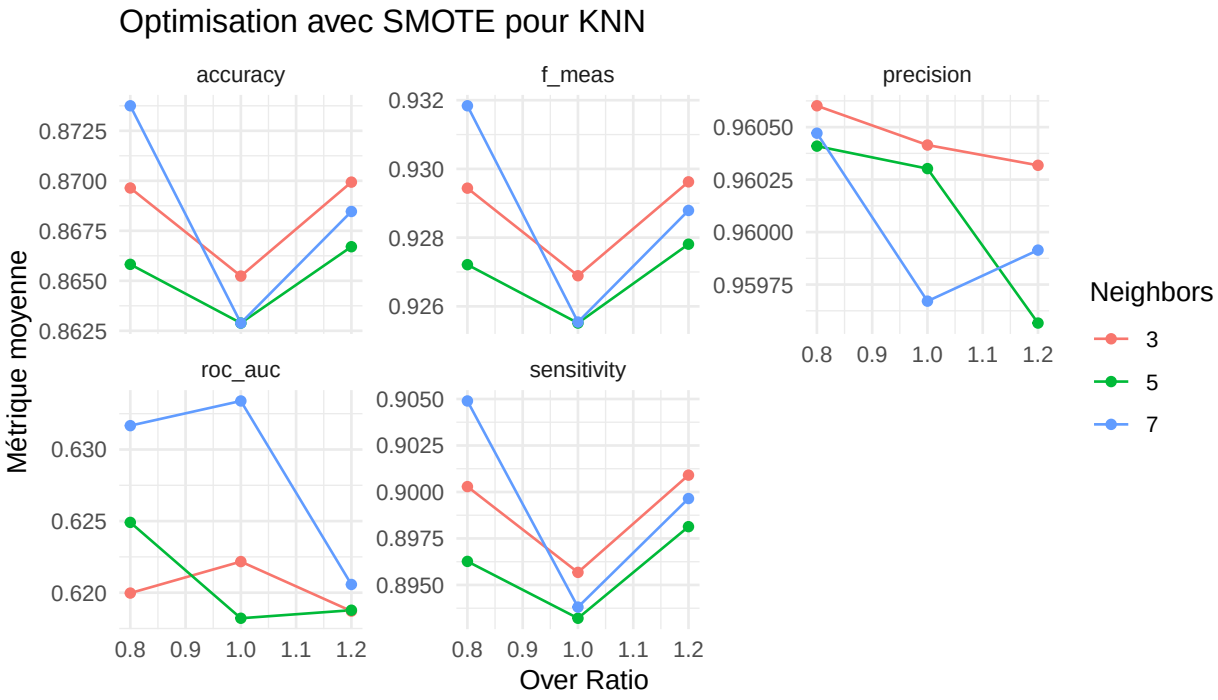


Table 23 – Meilleurs paramètres de SMOTE

over_ratio	neighbors	.config
0.8	**7**	**Preprocessor7_Model1**

Les graphiques illustrent l'impact des paramètres de suréchantillonnage SMOTE sur les performances du modèle KNN. De manière générale, un over_ratio de 1 semble dégrader légèrement les performances pour la majorité des métriques, en particulier accuracy, f_meas et sensitivity.

Le meilleur compromis entre précision, F1-score et capacité de discrimination est atteint avec 7 voisins et un over_ratio de 0.8, qui assure de bonnes valeurs pour toutes les métriques, notamment une précision maximale et une F1-score élevée. Ces paramètres seront donc privilégiés pour l'entraînement final du modèle KNN, car ils permettent un bon équilibre entre détection des cas positifs et robustesse globale du modèle.

11.2 Matrice de confusion

Table 24 – Matrice de confusion : prédiction AVC (KNN)

	Prédiction		
Réalité	**Absence AVC**	**Présence AVC**	**Total**
Absence AVC (0)	1452	160	**1612**
Présence AVC (1)	67	25	**92**
Total prédit	**1519**	**185**	**1704**

11.3 Tableau des métriques & AUC

Table 25 – Metrics

Métrique	Valeur (%)
Exactitude	86.68
Précision	13.51
Sensibilité	27.17
Spécificité	90.07
F1-score	18.05

Table 26 – Valeur de l'AUC pour le modèle KNN

Modèle	AUC..aire.sous.la.courbe.
KNN	0.634

12 Conclusion du modèle KNN

Le modèle KNN présente des performances globales modérées, avec un AUC de 0.634, ce qui reste relativement faible pour un modèle de classification, en particulier dans un contexte médical où la précision diagnostique est cruciale. Cette valeur, assez proche du seuil de 0.5, suggère une capacité discriminative limitée, à peine supérieure à ce qu'on obtiendrait avec une classification aléatoire.

L'analyse de la matrice de confusion révèle un modèle déséquilibré dans ses prédictions.

D'un côté, il excelle à identifier correctement les non-cas avec une spécificité élevée de 90.07% (1452 vrais négatifs sur 1612 cas réellement négatifs).

Cependant, le modèle présente une faiblesse majeure dans sa capacité à détecter les cas positifs d'AVC, avec une sensibilité de seulement 27.17%. En effet, cela signifie que sur 92 patients ayant réellement subi un AVC, le modèle n'en identifie correctement que 25, laissant 67 cas non détectés. Cette proportion élevée de faux négatifs est particulièrement problématique dans un contexte médical, où manquer un diagnostic d'AVC peut avoir des conséquences graves sur la santé du patient.

Par ailleurs, avec une précision de 13.51%, le modèle génère un nombre important de faux positifs. En effet, sur 185 patients identifiés comme à risque d'AVC, seuls 25 le sont réellement. Cela pourrait entraîner des examens complémentaires inutiles et une anxiété injustifiée chez les patients.

13 SVM Linéaire

La SVM linéaire permet de trouver un hyperplan qui sépare de manière optimale les données de différentes classes dans un espace de grande dimension. L'hyperplan est choisi de manière à maximiser la distance entre l'hyperplan et les points les plus proches de chaque classe.

Le paramètre à optimiser dans le modèle SVM linéaire est le coût, qui contrôle la pénalité associée aux erreurs de classification. Un coût plus élevé peut entraîner un modèle plus rigide, tandis qu'un coût plus faible peut rendre le modèle plus flexible, mais aussi plus susceptible à l'overfitting.

13.1 Recette & optimisation des hyperparamètres

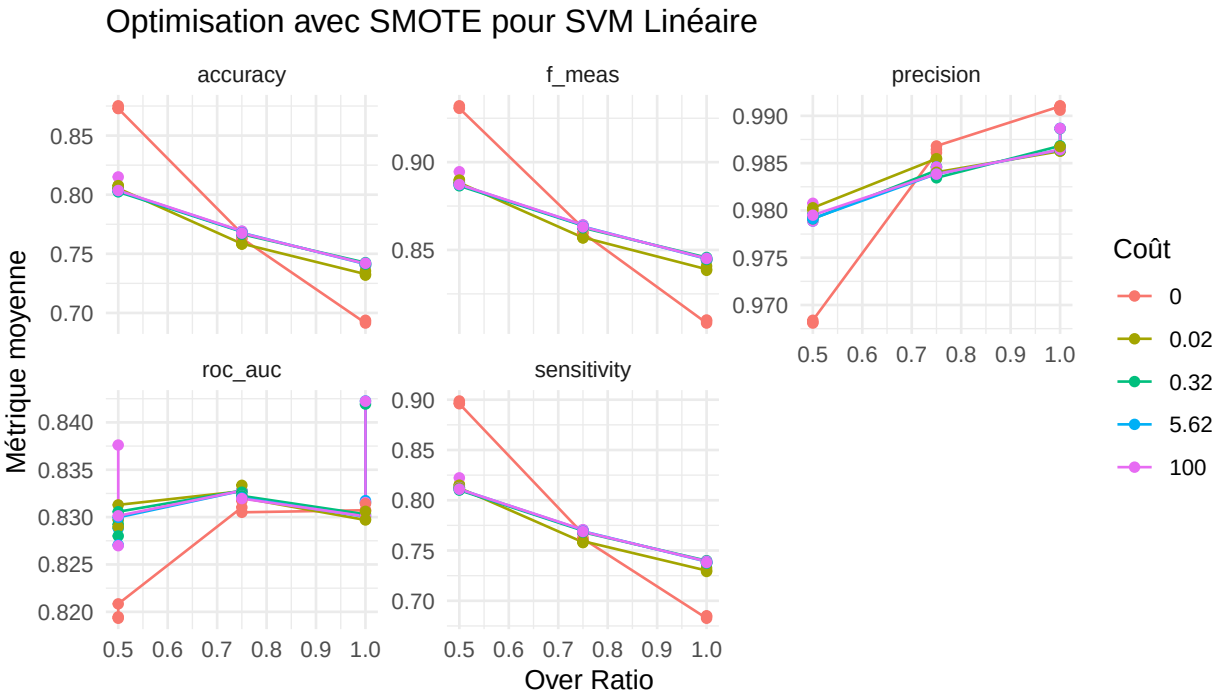


Table 27 – Meilleurs paramètres de SMOTE

cost	over_ratio	neighbors	.config
0.001	**0.5**	**3**	**Preprocessor1_Model1**

Les graphiques montrent l'évolution des métriques de performance du SVM linéaire en fonction du coût de pénalisation et du taux de suréchantillonnage (Over Ratio) appliqué via SMOTE.

Globalement, l'augmentation du coût n'apporte pas d'amélioration significative des performances. Les métriques restent relativement stables, voire légèrement dégradées lorsque le coût atteint des valeurs élevées (notamment à cost = 100), en particulier pour la métrique roc_auc.

Les meilleures performances en termes d'accuracy, f_meas et sensitivity sont observées pour des valeurs de coût faibles à modérées (autour de 0.32 à 5.62), ce qui suggère qu'un modèle moins rigide vis-à-vis des erreurs (pénalisation plus faible) est mieux adapté aux caractéristiques des données.

Ainsi, un coût de pénalisation faible permet d'obtenir un bon compromis entre performance et généralisation, tout en limitant les risques de surapprentissage. Ce paramètre sera donc retenu pour l'entraînement final du modèle SVM linéaire.

13.2 Matrice de confusion

Table 28 – Matrice de confusion : prédiction AVC (SVM linéaire)

	Prédiction		
Réalité	**Absence AVC**	**Présence AVC**	**Total**
Absence AVC (0)	1458	154	**1612**
Présence AVC (1)	53	39	**92**
Total prédit	**1511**	**193**	**1704**

13.3 Tableau des métriques & AUC

Table 29 – Metrics

Métrique	Valeur (%)
Exactitude	87.85
Précision	20.21
Sensibilité	42.39
Spécificité	90.45
F1-score	27.37

Table 30 – Valeur de l'AUC pour le modèle SVM linéaire

Modèle	AUC..aire.sous.la.courbe.
SVM LINEAIRE	0.8232

13.4 Conclusion Modèle SVM Linéaire

L'aire sous la courbe ROC du modèle SVM linéaire atteint 0.8232, ce qui indique une bonne capacité globale de discrimination entre les individus atteints et non atteints d'AVC, malgré des déséquilibres observés dans les performances par classe.

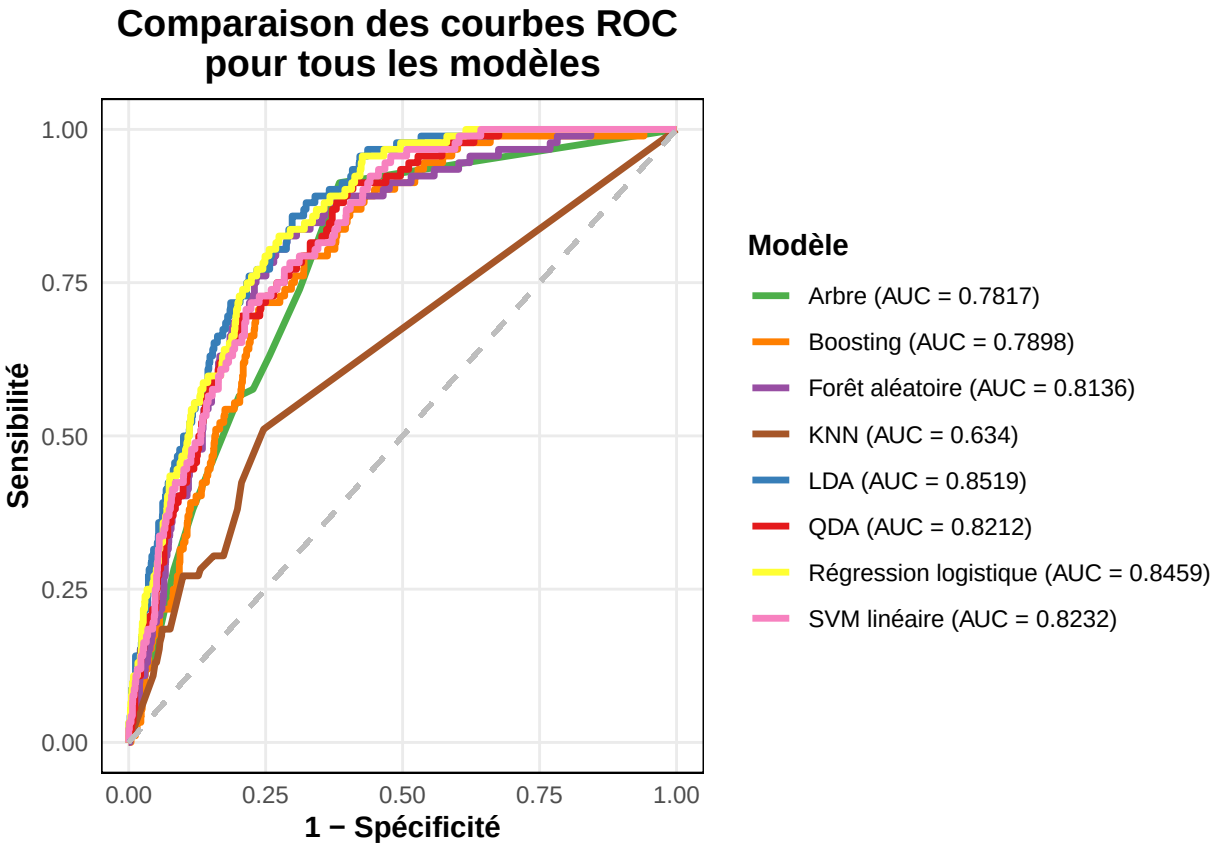
Le modèle SVM linéaire présente une bonne capacité à détecter les cas de non-AVC, comme en témoigne sa spécificité de 90.45 %. Cela signifie qu'il identifie correctement une grande majorité des individus ne présentant pas d'AVC. Toutefois, le modèle tend à surestimer les cas d'AVC, ce qui se traduit par un nombre non négligeable de faux positifs : parmi les 1612 individus réellement non atteints, 154 ont été prédits à tort comme présentant un AVC.

En revanche, sa capacité à détecter les véritables cas d'AVC reste limitée, avec une sensibilité de 42.39 %, soit 39 cas correctement identifiés sur 92 réels. Cette performance montre que le modèle parvient à capter une partie des individus à risque, mais laisse échapper une majorité des cas.

14 Comparaison des modèles

Voici un graphique présentant les courbes ROC des différents modèles.

On remarque que la courbe ROC du modèle des K plus proches voisins se distingue nettement des autres, en étant située plus bas.



15 Conclusion Générale

Etant donné que notre jeu de données contient principalement des variables qualitatives, nous avons choisi de ne pas retenir certains modèles comme la LDA, la QDA ou encore la régression logistique, qui sont moins adaptés dans ce contexte. Par ailleurs, à l'issue de notre comparaison, le modèle des K plus proches voisins (KNN) s'est révélé nettement moins performant, avec des scores très faibles sur l'ensemble des métriques évaluées. Il a donc été écarté de l'analyse finale.

En comparant les autres modèles, nous avons constaté que le SVM linéaire et le Boosting affichaient des résultats proches. Toutefois, le Boosting est plus complexe à mettre en œuvre et demande davantage de ressources, tant en termes de temps de calcul que de mémoire. Dans notre cas, le SVM linéaire constitue donc une solution plus efficace.

Enfin, dans le cadre de cette étude, notre priorité est de maximiser la détection des cas d'AVC, même si cela implique un certain nombre de fausses alertes. Il est en effet préférable d'alerter à tort que de manquer un vrai cas.